**Extract**

The data sets used in this project were downloaded from Kaggle.com. These sets originated from the YouTube API and include data from 10 countries. The data includes the top trending videos in each country for each day of the collection period.

For each of the countries there were two files:

A CSV file which houses the data for daily trending videos from November 2017 to June 2018.

A JSON file containing data on video categories.

**Transform**

An ERD was completed to enable visualization of the final tables and their relationships for ease of future analysis. To achieve the Second Normal Form, data was split into 3 tables.

1. Videos (Primary Key = video\_id) - Contains the video ID and title for each unique video
2. Countries (Primary Key = country\_code) - Contains the country code and country name for each country in the analysis
3. Video\_countries (Primary Key = video\_id, country\_code, trending\_date; Foreign Keys = video\_id, country\_code) - Contains data on trending date, view count, and comment count for the full set of trending videos

Data from the JSON files was converted to CSV to allow for inclusion in PostgreSQL if desired, but this project did not include these files in the database due to time considerations.

The video CSV files were transformed using the Pandas module in Python to work with the data in data frames.

* Formatted the CSV file for each country using data frames to create **VIDEO\_COUNTRIES** table
  + Eliminated all columns except (video\_id, title, views, comment\_count, and trending\_date)
  + Added a column to note the country code
  + Created a dictionary object to hold the clean data frame for each country
  + Concatenated all data frames by calling the dictionary items
  + Removed the title column (only after creating the **VIDEOS** table)
  + Dropped duplicate rows based on primary key columns after discovering their presence
  + Sent the final data frame to CSV
* Created the **VIDEOS** table containing one row for each video
  + Used the final data frame for the **VIDEO\_COUNTRIES** table
  + Dropped duplicate instances of video\_id
  + Eliminated all columns except (video\_id, title)
* Created the COUNTRIES table containing country codes and names
  + Looked up country codes and created table manually

Tables were created in a PostgreSQL database using PgAdmin 4.

**Load**

The data was loaded into the tables in PostgreSQL. PostgreSQL was chosen to allow for future updates and analysis, as well as the ability to assign relationships between the normalized table data. The database will ensure that, were API pulls to resume, no videos that are not present in the videos table are added to the larger video\_countries table.